

NVDA/P000502

**CLAIMS:**

**Claim 1 (Currently Amended)** A method for indexing and storing vertex data associated with the vertices that define ~~defining at least one primitive~~ neighboring primitives, comprising:

~~assigning a unique reference to each vertex defining the at least one primitive;~~

selecting a reference vertex;

identifying one-ring neighbor vertices of ~~[[each]]~~ the reference vertex;

~~assigning [[the]] a unique reference of each vertex to each of the one-ring neighbor vertices of each vertex; [[and]]~~

~~assigning a unique neighbor index to each of the one of the one-ring neighbor vertices of each vertex in a sequential order around the reference vertex; and~~

storing the neighboring primitives associated with the one-ring neighbors based on the assigned neighbor indexes.

**Claim 2 (Currently Amended):** A method for indexing as in Claim 1, wherein ~~[[a]]~~ each unique neighbor index includes an offset which is unique to each of the neighboring primitives to specify a consistent order of calculation during primitive processing.

**Claim 3 (Original)** A method for indexing as in Claim 2, wherein the offset is used to specify a consistent order of calculation for use during primitive processing.

**Claim 4 (Currently Amended):** A method for indexing as in Claim ~~[[1]]~~ 2, wherein the at least one primitive is a polygonal primitive.

**Claim 5 (Original):** A method for indexing as in Claim 4, wherein the polygonal primitive is a quadrilateral primitive.

**Claim 6 (Original):** A method for indexing as in Claim 4, wherein the polygonal primitive is a triangular primitive.

**Claim 7-8 (Cancelled):**

NVDA/P000502

Claim 9 (Original): A method for indexing as in Claim 1, wherein the at least one primitive defines a volume.

Claim 10 (Original): A method for indexing as in Claim 9, wherein the at least one primitive is a tetrahedron.

Claim 11 (Original): A method for indexing as in Claim 9, wherein the at least one primitive is a cube.

Claim 12 (Original): A method for indexing as in Claim 1, further comprising identifying an edge between a first vertex and a second vertex, the second vertex being a one-ring neighbor of the first vertex.

Claim 13 (Original): A method for indexing as in Claim 12, further comprising:  
assigning the unique reference of the first vertex to the edge; and  
assigning the unique neighbor index of the second vertex to the edge.

Claim 14 (Original): A method for indexing as in Claim 13, further comprising:  
assigning the unique reference of the second vertex to the edge; and  
assigning the unique neighbor index of the first vertex to the edge.

Claim 15 (Currently Amended): A method for indexing for data input to a graphics program, comprising:  
identifying a vertex;  
assigning a reference to the vertex to define a first reference vertex;  
identifying one-ring neighbor vertices of the vertex;  
assigning the reference to each of the one-ring neighbor vertices identified;  
assigning a neighbor index to one of the one-ring neighbor vertices identified;  
successively incrementing the neighbor index to provide incremented neighbor indices for assignment to the one-ring neighbor vertices remaining; and  
sequentially assigning one of the incremented neighbor indices to each of the one-ring neighbor vertices remaining, wherein the ordering of the one-ring neighbor vertices is user determined.

NVDA/P000502

**Claim 16-17 (Cancelled):**

**Claim 18 (Original):** A method for indexing as in Claim 15, further comprising:

totaling the one-ring neighbor vertices sharing an edge with the vertex to provide a total; and

indicating the total as a valence of the vertex.

**Claim 19 (Original):** A method for indexing as in Claim 15, wherein the neighbor index includes an offset.

**Claim 20 (Original):** A method for indexing as in Claim 15, wherein the vertex and at least a portion of the one-ring neighbors define a primitive.

**Claim 21 (Original):** A method for indexing as in Claim 15, further comprising:

assigning a second reference to one of the one-ring neighbor vertices to define a second reference vertex;

identifying one-ring neighbor vertices of the second reference vertex, the one-ring neighbor vertices including the first reference vertex;

assigning the second reference to each of the one-ring neighbor vertices of the second reference vertex identified;

assigning an additional neighbor index to one of the one-ring neighbor vertices of the second reference vertex identified;

successively incrementing the additional neighbor index to provide incremented additional neighbor indices for assignment to the one-ring neighbor vertices of the second reference vertex; and

sequentially assigning one of the incremented additional neighbor indices to each of the one-ring neighbor vertices of the second reference vertex remaining.

**Claim 22 (Original):** A method for indexing as in Claim 21, wherein the first reference vertex may be referenced using the first reference or using the second reference and an additional neighbor index assigned.

**Claim 23 (Original):** A method for indexing as in Claim 22, wherein data corresponding to the first reference vertex is stored in a portion of memory accessed using the first

NVDA/P000502

reference.

Claim 24 (Original): A method for indexing as in Claim 23, wherein data corresponding to the first reference vertex stored in the portion of memory is accessed using the second reference and the additional neighbor index assigned.

Claim 25 (Original): A method for indexing as in Claim 21, wherein the second reference vertex may be referenced using the first reference and a neighbor index assigned or using the second reference.

Claim 26 (New): A method as in claim 1 wherein edges are referenced in relation to a selected vertex of an originating primitive and one neighbor vertex of the selected

Claim 27 (New): A method for indexing vertex data defining at least one primitive, comprising:

- assigning a unique reference to each vertex defining the at least one primitive;

- identifying one-ring neighbor vertices of each vertex;

- assigning the unique reference to each of the one-ring neighbor vertices of each vertex; and

- assigning a unique neighbor index to each of the one of the one-ring neighbor vertices of each vertex. wherein a unique neighbor index includes an offset.

Claim 28 (New): The method of claim 27 wherein the offset is determined by the number of one-ring neighbor vertices to a reference vertex.

Claim 29 (New): The method of claim 28 wherein the offset determines the sequence with which the neighboring primitives are processed.